

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (original) A method of scheduling an event, the event involving a plurality of resources, the method comprising performing a process in respect of each resource, the process comprising identifying a slot time corresponding to a time at which the resource is available ; and creating a software component corresponding to the identified slot time, wherein the software component comprises communicating means arranged to communicate with other like software components, and storage arranged to store data in respect of the resource corresponding to the software component and data in respect of the identified slot time; and wherein each software component so created communicates with another like software component in order to identify a time for the event that satisfies a predetermined criterion.

2. (original) A method according to claim 1, in which identifying a time for the event includes, for each slot time

accessing the slot time data stored by the software components in order to identify software components corresponding thereto;

using stored data in respect of the resources corresponding to the identified software components to evaluate the suitability of the slot time for the event;

and

selecting a time for the event in accordance with the evaluated suitabilities.

3. (currently amended) A method according to claim 1 ~~or claim 2~~, in which the process includes accessing a schedule associated with the said resource and retrieving data indicative of the availability of the resource.

4. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1, in which, for each slot time, the method includes storing data in respect of the resources corresponding to the identified software components.

5. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1, in which, for each slot time, the method includes comparing the data in respect of the resources corresponding to the identified software components with data in respect of the plurality of resources involved in the event, so as to identify those resources for which there is no software component,

sending a message to the identified resource, the message including a request for preference information in respect of the slot time,

receiving said preference information and

if the preference information indicates that the resource is available at that slot time, updating the data in respect of the resources corresponding to the identified software components.

6. (currently amended) A method according to ~~any one of claims 2 to 5~~ claim 2, in which the step of evaluating the suitability of a slot time comprises calculating the number of resources corresponding to the identified software components corresponding to the slot time and comparing the calculated number with a specified criterion.

7. (currently amended) A method according to ~~claim 5 or claim 6~~, in which the step of evaluating the suitability of a slot time comprises comparing the data in respect of the resources corresponding to the identified software components with data in respect of one or more resources whose involvement with the event is essential.

8. (currently amended) A method according to claim 6 ~~or claim 7~~, in which, if the preference information indicates that the resource is unavailable at that slot time, the method includes incorporating this preference information in the suitability evaluation.

9. (currently amended) A method according to ~~any one of claims 2 to 8~~ claim 2, including notifying the resources corresponding to the identified software components corresponding to the slot time of the selected slot time.

10. (original) A method according to claim 9, in which, in the event that the selected time becomes unavailable, the method includes sending a message indicative of a change in status of the selected slot time to the notified resources, and reviewing the stored suitability to select a replacement slot time.

11. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1, in which the process includes storing a time of creation of the software component corresponding thereto.

12. (original) A method according to claim 11, in which said step of identifying a slot time corresponding to a time at which the resource is available includes

identifying which of the identified software components was created first;

adding data in respect of resources corresponding to the other software components to the storage of the identified first created software component and

deleting the other software components.

13. (currently amended) A method according to ~~any one of the preceding claims~~ claim 1, including selecting an initiating resource, the initiating resource being one of the plurality, performing the process in respect of the initiating resource, sending event details to the others of the plurality of resources, and performing the process in respect of the others of the plurality of resources.

14. (original) A software component for use in selecting a time for an event, wherein the event involves a plurality of resources, the software component comprising

communicating means arranged to communicate with other like software components and

storage arranged to store data in respect of a resource corresponding to the software component, the data including a time at which the resource is available for the event,

the software component being arranged, in use, to communicate with other software components to identify those software components storing data relating to the same time, and, for any software components so identified, the software component is arranged to store data relating to the identified software components in the storage and evaluate the suitability of the time for the event.

15. (original) A software component according to claim 14, wherein the storage includes data identifying a time of creation of the software component, the software component further being operable to access times of creation corresponding to the identified software components, and, if the time of creation corresponding to the software component is later than that corresponding to any one of the identified software components, the software component is operable to destroy itself.

16. (currently amended) A diary system for use in scheduling an event on behalf of a user, comprising

schedule querying means arranged to identify one or more times at which the user is available for the event,

software component creating means arranged to create a software component according to claim 14 or 15, wherein the data stored in the storage thereof includes one of the identified times, and

schedule updating means arranged to receive, from the, or another such software component, data indicative of a time at which the event is to be scheduled, and to update the schedule in accordance therewith.

17. (original) A diary system according to claim 16, wherein, in the event that the software component receives data indicative of a failed scheduling attempt, the software component creating means is arranged to create a further software component corresponding to one of the other times at which the user is available for the event.

18. (original) A method of selecting a time for an event, the event involving a plurality of resources, the method comprising the steps of

performing a process in respect of each resource, the process comprising

identifying a slot time corresponding to a time at which the resource is available;

creating a software component corresponding to the identified slot, the software component comprising communicating means arranged to communicate with other like software components and storage arranged to store data in respect of the resource corresponding to the software component and data in respect of the identified slot time; for each slot time:

accessing the slot time data stored by the software components in order to identify software components corresponding thereto;

using stored data in respect of the resources corresponding to the identified software components to evaluate the suitability of the slot time for the event; and

selecting a time for the event in accordance with the evaluated suitabilities.

19. (original) A method of distributing a plurality of tasks between a plurality of resources, comprising

performing a process in respect of each resource, the process comprising

identifying a processing capability of the resource; and

creating a software component corresponding to the identified capability, wherein the software component comprises communicating means arranged to communicate with other like software components and storage arranged to store data in respect of the resource corresponding to the software component, the data including the identified capability;

wherein each software component so created is operable to communicate with another like software component in order to identify distribution of tasks that satisfies a predetermined criterion.

20. (original) A method according to claim 19, in which the process includes selecting at least one of the plurality of tasks in accordance with the identified processing capability and storing the same in the storage of the created software component.

21. (original) A method according to claim 20, in which said predetermined criterion includes a condition that each of the plurality of tasks has been selected by at least one software component.

22. (currently amended) A method according to claim 20 ~~or claim 21~~, in which said predetermined criterion includes a cost associated with the selection, so that the step of identifying distribution of tasks include evaluating the cost associated therewith.

23. (currently amended) A method according to ~~any one of claims 20 to 22~~ claim 20, in which, in the event that one of the plurality of tasks has been selected by more than one software component, the method includes comparing utilisation of processing capabilities between resources, and allocating said one task to whichever resource has the most efficient processing capability utilisation.